Client-side Wi-Fi Roaming via Radio Environment Maps: Channel Charting vs. Fingerprinting

Wendlasida Ouedraogo
SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris,
Palaiseau, France

Andrea Araldo
SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris,
Palaiseau, France

Lucas Bréhon–Grataloup
SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris,
Palaiseau, France

Nadjib Achir
INRIA, France

Antoine Lavignotte
SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris,
Palaiseau, France

***Abstract*—Default roaming procedures heavily rely on static parameters, such as waiting for the serving Access Point (AP) signal to drop below a preset threshold. Although simple, this scheme fails to adapt to the changing radio conditions an end device may encounter as it moves, leading to excessive or sticky roaming. This raises the following question: *how can roaming parameters be dynamically adapted to the local radio environment?* To address this problem, we propose a two-phase client-side framework for adaptive Wi-Fi roaming based on radio environment maps (REM). In the offline phase, we use raw Channel State Information (CSI) measurements collected at the user device to build a REM using unsupervised CSI representation techniques, namely fingerprinting (FP) and channel charting (CC). Neither requires explicit location labels nor incurs communication overhead, as they rely on passive CSI sensing. The resulting REM is then partitioned into clusters corresponding to physical areas, and each cluster is associated with zone-specific roaming parameters, namely hysteresis threshold and margin, derived from the collected RSSI statistics. In the online phase, newly collected CSI are continuously mapped within the REM to estimate the current cluster and dynamically select the appropriate roaming parameters. Our proposition can fully run on the device side without an external localization solution or infrastructure support, making it compatible with the user-side aspect of the Wi-Fi AP selection scheme. We show that our approach reduces unnecessary handovers while maintaining signal quality similar to a static-parameter baseline.**

***Index Terms*—Wi-Fi, Unsupervised Learning, Roaming, Channel Charting, Fingerprinting**

I. Introduction

Wi-Fi has emerged as the predominant indoor radio access technology, meeting end devices' requirements (e.g., coverage, data rate) while remaining easy to deploy and manage [1]. Since its introduction, Wi-Fi has undergone several improvements, including improvements to the roaming procedure (e.g., through amendments such as 802.11k/v/r [2]–[4]). Roaming is the mechanism by which an end device, which we call a client, maintains its connection while moving by switching between available Wi-Fi APs. The decision of when and where to switch is generally left to the client, who relies, by default, on purely heuristic-based approaches that use static parameters (e.g., RSSI hysteresis thresholds and margin) to select between candidate APs. This is an easily deployable scheme and when properly parameterized, it allows the client to avoid erratically switching between APs with similar signal quality (**the ping-pong problem**) or to remain connected to a suboptimally performing AP (**the sticky client problem**).

However, by default, these parameters are static and pre-configured, with no guarantee that the values are well set. Indeed, such values depend on the operating system or device model [5], [6], and are not tuned by default. Typically the device roams when a candidate AP exceeds the current AP's RSSI threshold τ by a margin Δ. Those parameters are never adapted to changing radio conditions, leaving the client prone to ping-pong and sticky client issues [7].

A solution explored in literature is to make those parameters adaptive: dynamically changing depending on contextual information, such as local radio conditions, throughput history, user location, or mobility. In particular, for location-aware adaptive solutions, two challenges have to be resolved: *(i)* how location information is obtained, and *(ii)* what AP selection policy must subsequently be applied. However, most of these works assume location information is already available [7], [8], addressing challenge *(ii)* but not *(i)*. Our work targets both, resorting to the client's CSI passive measurements, with no explicit location information nor infrastructure coordination, while proposing a simple yet efficient *(ii)* AP policy.

II. Background and Motivation

A. Roaming Procedure

Roaming allows a client to maintain its connectivity by switching seamlessly between available APs as it moves. In general, any client receives either frames from its active connection or from beacon frames broadcast by APs on their operating channel. From those frames, the client can estimate metrics (e.g., RSSI and SNR, throughput). Doing so, any client can continuously evaluate its current connection and track neighboring AP performance.

Roaming procedure is triggered when the current AP does not perform well enough : (e.g., RSSI drops below a preset threshold) [5]. The client, then seeks candidate APs through

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

both passive and active scanning. Passive scanning consists of listening for beacons, while active scanning involves sending probe requests, to which APs respond with probe responses, allowing the client to assess candidate networks' performance. This scanning phase is improved by the 802.11k [2] amendment, which enables an AP to provide a client with information about neighboring APs. How exactly the client evaluates those APs' performance is vendor-specific. For instance, Android scores candidate APs based on multiple criteria, such as throughput, internet access, and, in addition, signal strength [6]. Lastly, the client associates with the selected AP, if it differs from the current one.

B. Channel Charting

Channel Charting (CC) was introduced in [9], as an unsupervised framework that learns a low-dimensional representation of CSI. This representation, the channel chart, preserves local geometry and can be used as a proxy for user location. CC has been successfully applied to tasks such as beam SNR prediction [10] or proximity detection [11]. State-of-the-art methods involve neural network-based approaches with metric-learning techniques, such as triplet or contrastive losses, to map high-dimensional CSI features onto the channel chart. Other techniques such as Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP), have been used in works like [9], [10], [12].

C. Fingerprinting

Fingerprinting is a well-established framework for wireless localization that relies on the unique spatial properties of the radio signal [13]. It consists of finding fingerprints, CSI-based signatures that are similar within the same physical area. FP can be applied as a supervised technique, where fingerprints are mapped to their known locations using true ground-truth position data (e.g., GNSS) as labels [13]. However, in this work, we employ FP in an unsupervised approach, treating fingerprints as signatures of physical zones without mapping them to their true locations. Techniques used to identify fingerprints range from DBSCAN (density-based clustering) to K-means clustering [14].

III. Proposed Solution: Unsupervised Learning for Adaptive AP Selection

We consider a 2D indoor environment (e.g., an office) with N APs installed at fixed locations and a mobile client moving through the environment.

We propose a data-driven approach subdivided into two stages. An offline learning stage in which the device learns its radio environment using passively acquired signal strength and CSI, and adapts hysteresis parameters for application in the second online stage to newly acquired CSI. We focus on two specific parameters: τ and Δ. These two steps are described in Fig 1 and further detailed below.

We assume a stationary radio environment between the offline learning phase and online deployment, so that changes in the environment (e.g., people moving) are not expected to affect the radio conditions.

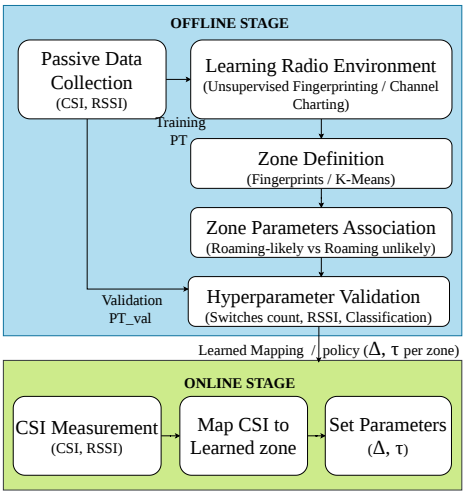


Fig. 1: Proposed two-phase pipeline.

A. Offline Stage

The offline stage consists of five phases:

1) **Data collection**. The user collects raw CSI and RSSI measurements by solely passive sensing of ambient wireless signals. We assume that the client collects CSI across PT trajectories, which is subsequently split into PT_{train} and PT_{val}.
2) **Learning REM** on recorded PT_{train} data through FP and CC. Both involve a preprocessing step that consists of applying Hampel and Savitzky–Golay filters, then extracting the amplitude of CSI vectors. Learning the radio environment depends on the framework;
   - *Fingerprinting.* We use K-Means to cluster feature vectors into n^{fp}_{clusters} distinct fingerprints.
   - *Channel charting.* We apply UMAP [15] dimensionality reduction to the feature vectors to obtain a two-dimensional channel chart.
3) **Zone definition.**
   - *For fingerprinting*, this step is straightforward. We consider each cluster as the signature of a physical zone.
   - For *Channel charting*. We use K-Means to subdivide the obtained chart into $n^{cc}_{clusters}$ clusters.

   The key assumption supporting the idea that the learned clusters correspond to actual physical zones is the same supporting channel charting and fingerprinting frameworks, i.e., feature vectors from physically close locations are similar (verified in Sec. V-C).
4) **Zone Parameters Association**. Each discovered zone is associated with parameters. The main objective is to reduce unnecessary switches. For this, we defined two types of zones based on RSSI average gap between the 2 best performing APs (that we denote as AP_i and AP_j): **roaming likely** zones, the n_{rl} zones where this gap is the smallest, and **roaming unlikely** zones, the remaining zones, where one AP is dominant. The association is then as follows:

- In roaming likely zones, we forbid switches. Practically, it consists in setting an arbitrarily high τ.
- In roaming unlikely zones, a switch is only necessary if the client is attached to the APs with the lowest perceived RSSI. We therefore keep a hysteresis margin $\Delta = 8db$ (like default hysteresis) and a threshold $\tau = (\overline{RSSI_i} + \overline{RSSI_j})/2$, where $\overline{RSSI_k}$ is the mean RSSI of AP k over the samples within the cluster (zone). We assume that the RSSI gap between APs is stable within a zone during both the offline and online phases.

5) **Hyperparameters validation** on PT_{val} recorded trajectories data. We validate $n^{fp}_{clusters}$, $n^{cc}_{clusters}$, and n_{rl} for each framework through grid-searching over candidate values. Note that this validation step is still offline, performed using logged data, with no parameters modified yet.

B. Online Stage

In this phase, the client continuously maps new CSI measurements onto the learned REM and identifies the client's current zone, in order to enforce the learned policy, as shown in Fig. 1. Similarly to the offline stage, each new raw CSI vector is first preprocessed, and its amplitude extracted, then subsequently mapped into a roaming likely or unlikely zone and the hysteresis parameters are finally applied.

IV. Dataset And Experiments

A. Experimental Setup

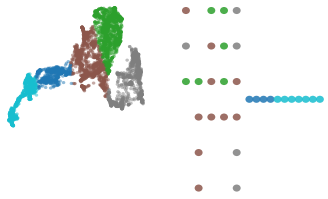
Our experimental environment is a single floor equipped with 2 APs and 2 additional devices, used only to generate traffic. We build a dataset of RSSI and CSI measurements of 802.11ac frames, over a 80MHz band collected using a Raspberry Pi (RPi). Our experimental environment is subdivided into 69 reference points (RPs), at which the RPi sequentially records measurements without transmitting itself. CSI extraction relies on the Nexmon_CSI tool [16]. We conduct multiple measurement campaigns during which the RPi passively measures ambient signals at each RP sequentially.

Realistic Path Generation. In both the online and offline phases, we generate realistic trajectories that reflect the client's actual mobility; each trajectory is a list of contiguous RPs. We consider that the client's collected measurements are drawn from campaign data: at each RP_i, the client's CSI and RSSI measurements are sampled (at a rate of $f_s = 10$ samples per step) from the given campaign's real measurements at the same location (RP_i). This follows a similar approach as in the training phase of [17]. This approach allows for considering multiple mobility scenarios while reducing the need for labor-intensive measurements.

V. Results

A. Key Metrics

We consider three roaming performance metrics. They are all first averaged over each trajectory, then over all considered trajectories. **Switches** count. The number of switches between APs. **Avg. RSSI**. The RSSI experienced by the client. **Sub-optimal time (%)**: The proportion of time steps within a trajectory at which the serving AP offers a lower RSSI than the alternative AP.



(a) CC (left), Physical Map (right). obtained by sensing AP_1



(b) FP (Physical Map) (c) PHY (Physical Map)

Fig. 2: $n^{fp}_{clusters} = n^{cc}_{clusters} = 5$. Each cluster is associated with a different color. On the physical map, each RP is colored according to its most frequent cluster assignment (majority vote over samples at that RP). The clusters discovered are different across techniques (CC, FP, PHY).

B. Baselines

We compare our approach with 4 baselines:

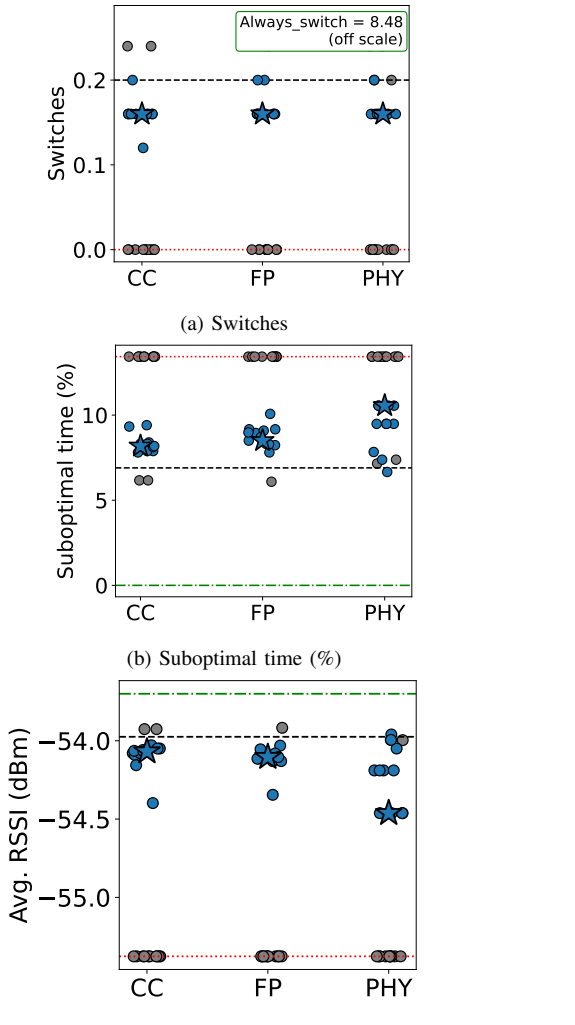
- A **simple hysteresis** with static parameters ($\tau = -70dBm$, $\Delta = 8dB$) which is the default one deployed on clients.
- **A physical location aware(PHY)**: for which we suppose the physical location is known. In other words, the roaming policy is learned from the true physical positions (x, y) clustered via K-means, following the same principle as our approach.
- **Always-switch:** switches to the best AP at each step.
- **No-switch:** keeps connected to initial AP attachment.

We compare these baselines against our proposed fingerprinting-based and a CC-based approach.

C. Learning And Online Stage

For the following results, we consider campaign 1 data for offline learning and campaign 2 for online test, following the process described in Sec. IV-A to generate realistic trajectories (PT_{train} for training and PT_{test} for test) dataset collection. Campaigns 1 and 2 are a few hours apart (3h).

Learning Radio Map. Fig. 2 shows the clusters discovered during the learning stage mapped to their physical location, in addition to the channel chart obtained. In the PHY approach, we observe that clusters perfectly correspond to actual physical areas, as expected, since this approach straightforwardly clusters (x, y) locations. For the FP-based and CC-based approaches, discovered clusters are more mixed in the physical map, particularly in the square room.



(a) Switches

(b) Suboptimal time (%)

(c) Avg. RSSI (dBm)

Fig. 3: Configurations performances online. Blue points satisfy validation filtering; gray points do not.

Hyperparameters validation. We sweep $n^{cc}_{clusters}$, $n^{fp}_{clusters} \in \{1, \ldots, 6\}$ and $n_{rl} \in \{1, \ldots, 4\}$. The chosen parameter configuration must satisfy the following criteria:

- Achieve fewer switches than the simple hysteresis, which uses static parameters (currently the one deployed).
- Achieve at least one switch to eliminate the cases where all zones are classified as roaming likely.

We first apply grid search offline and select configurations based on validation step criteria. We show the performance of both selected configurations and failed ones in the online phase in Fig. 3. We show configuration performances across the 3 approaches, FP, CC based, compared with baselines. The selected configurations in blue consistently achieve fewer switches than the simple hysteresis baseline without falling into the zero-switch policy. However, the cost is more suboptimal time, even if it doesn't translate into a significant loss in average RSSI, which is consistently the closest to the simple hysteresis and the always-switch policy.

VI. Conclusion and Future Work

In this paper, we leverage both FP and CC frameworks for passive signal measurement to learn a REM and derive an adaptive roaming strategy. We show that in a frequently visited environment, such as an office, our method, which relies exclusively on passive sensing, can replace the default roaming strategy based on static parameters and active scanning. Experimental results show that the proposed method reduces the number of AP switches while maintaining satisfactory signal quality. Several limitations remain, and ongoing work aims to address them: considering more APs, additional metrics such as throughput and congestion, continuously moving device as well as a full characterization of the cost of learning (memory usage or energy consumption).

Acknowledgment

This work was supported by the Chaire Réseaux du futur pour les Services de demain.

References

[1] Wi-Fi Alliance. The prominent role of Wi-Fi in powering next-generation use cases. [Online]. Available: https://www.wi-fi.org/beacon/the-beacon/prominent-role-of-wi-fi-in-powering-next-generation-use-cases
[2] *IEEE 802.11k-2008*, IEEE Std., 2008.
[3] *IEEE 802.11v-2011*, IEEE Std., 2011.
[4] *IEEE 802.11r-2008*, IEEE Std., 2008.
[5] Apple Inc. (2024) Wi-fi roaming support in apple devices. Accessed: 2026-06-16. [Online]. Available: https://support.apple.com/guide/deployment/wi-fi-roaming-support-dep98f116c0f
[6] Android Open Source Project. (2026) Wi-fi network selection. Accessed: 2026-06-16. [Online]. Available: https://source.android.com/docs/core/connect/wifi-network-selection
[7] K. K. Thangadorai, K. M. Sivalingam, H. P. Gupta, and M. R. Kanagarathinam, "Stickyless: An intelligent method for solving sticky client problem in wi-fi networks," in *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, 2024, pp. 1–6.
[8] K.-L. Yap, Y.-W. Chong, and W. Liu, "Enhanced handover mechanism using mobility prediction in wireless networks," *PLOS ONE*, vol. 15, no. 1, p. e0227982, Jan. 2020.
[9] C. Studer, S. Medjkouh, E. Gonultaş, T. Goldstein, and O. Tirkkonen, "Channel Charting: Locating Users Within the Radio Environment Using Channel State Information," vol. 6, pp. 47 682–47 698.
[10] P. Kazemi, H. Al-Tous, T. Ponnada, C. Studer, and O. Tirkkonen, "Beam SNR Prediction Using Channel Charting," vol. 72, no. 10.
[11] P. Ferrand, M. Guillaud, C. Studer, and O. Tirkkonen, "Wireless channel charting: Theory, practice, and applications," *IEEE Communications Magazine*, vol. 61, no. 6, pp. 124–130, 2023.
[12] P. Ferrand, A. Decurninge, L. G. Ordoñez, and M. Guillaud, "Triplet-based wireless channel charting: Architecture and experiments," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, 2021.
[13] X. Wang, L. Gao, and S. Mao, "CSI Phase Fingerprinting for Indoor Localization With a Deep Learning Approach," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1113–1123, 2016.
[14] J. Torres-Sospedra, D. Quezada-Gaibor, G. M. Mendoza-Silva, J. Nurmi, and J. Huerta, "New cluster selection and fine-grained search for k-means clustering and wi-fi fingerprinting," in *ICL-GNSS*, 2020, pp. 1–6.
[15] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2020.
[16] F. Gringoli, M. Schulz, J. Link, and M. Hollick, "Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets," in *International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, 2019, p. 21–28.
[17] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate rssi indoor localization," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 639–10 651, 2019.